AFRL-IF-RS-TR-2005-363
Final Technical Report
October 2005

REAL-TIME DECISION SUPPORT FOR COURSE OF ACTION/ENEMY COURSE OF ACTION (COA/ECOA) ANALYSIS

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-363 has been reviewed and is approved for publication

APPROVED: /s/

DUANE GILMOUR
Project Engineer

FOR THE DIRECTOR: /s/

JAMES A. COLLINS, Deputy Chief
Advanced Computing Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
OCTOBER 2005

3. REPORT TYPE AND DATES COVERED
Final May 05 – Aug 05

4. TITLE AND SUBTITLE
REAL-TIME DECISION SUPPORT FOR COURSE OF ACTION/ENEMY COURSE OF ACTION (COA/ECOA) ANALYSIS

5. FUNDING NUMBERS
C - FA8750-05-1-0234
PE - 62702F
PR - 4519
TA - ZA
WU - NG

6. AUTHOR(S)
Zhongfei Zhang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Zhongfei Zhang
201 Evergreen St. Apt 4-3B
Vestal New York 13650-0275

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFTC
26 Electronic Parkway
Rome New York 13441-4514

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2005-363

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Duane Gilmour/IFTC/(315) 330-3550/ Duane.Gilmour@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*
This report documents the research on course of action (COA) analysis in alignment with commander's intent. A military mission requires that a series of COAs be generated, developed, analyzed and executed. For each mission, thousands of COAs may be automatically generated but only those in alignment with Commander's Intent are worth investigating. The challenge is that given a specific pair of commander's intent and COA, there is always a semantic gap: the two not only differ syntactically, but also semantically. In this research, we have made two specific contributions towards developing a solution to this problem. First, we have discovered that the classic symbolic reasoning does not work in developing such a solution, as the semantics involved are always fuzzy and inexact. Second, under the assumptions that both the commander's intent and the COAs are represented in a low level in a semantic hierarchy (such that there is a syntax to represent them in terms of languages), we have developed a specific solution as a method to identify whether a specific pair of commander's intent and COA is in alignment and if not, how far they divert from each other. We have done proof-of-concept testing on a small, hand-crafted ontology.

14. SUBJECT TERMS
Commander’s intent, Course of Action, Semantic Inference, COA Analysis, Fuzzy Logic

15. NUMBER OF PAGES
18

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

Table of Contents

List of Figures

List of Tables

1.0 Executive Summary

This report documents the research on course of action (COA) analysis in alignment with the commander's intent. A military mission requires that a series of COAs be generated, developed, and executed. For each mission, thousands of COAs may be generated. Therefore, before actual analysis and execution of a COA, it is extremely important to determine whether this particular COA is in alignment with the Commander's Intent for this mission, such that the limited resources may be allocated to execute those COAs that are determined to be in alignment with the commander's intent. This capability plays a critical role in dominating a battlefield and consequently in succeeding in a military campaign. The challenge of this problem is that given a specific pair of commander's intent and COA, there is always a *semantic gap*: the two not only typically differ syntactically, but more importantly also semantically. In this summer research, we have made two specific contributions towards developing a solution to this problem. First, we have discovered that the classic symbolic reasoning does not work in developing such a solution, as the semantics involved are always fuzzy and inexact. Second, under the assumptions that both the commander's intent and the COAs are represented in a low level in a semantic hierarchy (such that there is a syntax to represent them in terms of languages), we have developed a specific solution as a method to identify whether a specific pair of commander's intent and COA is in alignment and if not, how far they divert from each other. This specific method is called CAFSIN, standing for COA analysis based on fuzzified semantic inference. We have done the proof-of-concept testing for CAFSIN on a small, hand-crafted ontology.

2.0 Introduction

In a military campaign, in order to accomplish a military mission, it is typically required that a series of courses of actions (COAs) be generated, developed, and executed. For a typical military mission, thousands of COAs may be generated. Clearly, it is neither possible nor necessary to execute all of the COAs. Instead, before the execution of a COA, we need to determine whether a particular COA is in alignment with the commander's intent for the mission. This capability is extremely important and plays a critical role in dominating the battlefield and consequently succeeding in the campaign. This research addresses the investigation of developing this capability.

There are two fundamental issues in addressing COA analysis. The first is the suitability analysis. A COA is suitable if it is in alignment with the commander's intent. The second is the feasibility analysis. A COA is feasible if it can be achieved with the given resources. The first issue relates to the semantic inference on whether the COA matches the commander's intent while the second issue relates to the COA scheduling and sequencing given whatever resources available.

A commander's intent is defined in terms of the goal and the end state. The goal is what the military campaign is expected to achieve. The end state is what the conditions are expected after the military campaign is over. Due to the existence of a typical military administrative hierarchy in command of a specific military campaign, a commander's intent may also be represented in different levels of a hierarchy, from the strategic level through the operational level to the tactical level.

The strategic level of the commander's intent refers to a high level commander's intent, such as the president's intent. An example of the strategic level commander's intent is shown in Figure 1.

```
We will liberate Orangeland, restore
power and control to her rightful
government, and then punish the
aggressor nation for its unlawful attack
and occupation by significantly reducing
his ability to wage war such that he is
no longer a regional threat.
```

Figure 1: An example of a strategic level commander's intent.

The operational level commander's intent refers to the actual execution commander's intent, i.e., the intent of the commander in charge of the specific military campaign. At this level, the commander's intent may be represented into several aspects, such as end state, purpose, method, and risk. An example of an operational level commander's intent is shown in Figure 2.

```
End state: a. freedom to operate forces
starting with pre-deployment activities;
b. no Weapons of Mass Destruction
(WMD)/Theater Ballistic Missile (TBM) or
terrorist threat to region/US
Purpose: regional stability and US
security
Method: Global Strike Task Force initial
strikes followed by Air and Space
Epeditionary Task Force/Carrier Battle
Group persistence forces
Risk: low to US forces; medium for
collateral damage
```

Figure 2: An example of an operational level commander's intent.

The tactical level of commander's intent refers to the specific objectives that the staff of the commander in charge of the military campaign has outlined in terms of the operational level commander's intent. An example of the tactical level commander's intent is shown in Figure 3.

```
Disrupt enemy TBM Command & Control (C2)
systems
```

Figure 3: An example of the tactical level commander's intent.

On the other hand, a COA actually represents a specific possible option in order to achieve a military mission, and therefore, it may also be represented in a hierarchy at different levels of execution. For example, a higher level COA may be “attack WMD power and TBM power” while a lower level COA may be “move FA-18 at speed 500 through route 21”. Consequently, a COA may consist of several lower level granularity COAs in sequence.

This research only addresses the suitability issue of COA analysis. In other words, given a commander’s intent and a COA, the problem is to determine whether the COA is in alignment with the commander’s intent, and if not, how far the COA diverts from the commander’s intent.

The challenge is that typically there are always semantic uncertainty and fuzziness for both commander’s intent and COAs. This semantic uncertainty and fuzziness demand that not only natural language be correctly understood, but also the semantic meaning of each word be correctly understood given the different context in different application. For example, referring to Figure 1, what do “control” and “ability significantly reduced” exactly mean? Due to this semantic uncertainty and fuzziness, there is a semantic gap between the commander’s intent and a COA; the challenge to developing a solution to this problem is to filling in this semantic gap.

In order to address the semantic uncertainty and fuzziness, we propose a fuzzified approach to semantic inference for COA analysis, called CAFSIN. We have demonstrated the effectiveness of CAFSIN method through preliminary testing and evaluations.

This report is organized as follows. After this introductory section, we briefly review the related work in the literature. Then we give detailed presentation of CAFSIN method, and report the preliminary testing and evaluations, before the report is concluded.

3.0 Related Work

COA analysis has received attention in recent military campaign research for years. Chandrasekaran and Josephson studied COA analysis through computer-generated forces in simulation using cognitive modeling [Chandrasekaran]. Based on individual cases, they used the cognitive modeling to attempt to develop a generalized strategy for COA analysis using simulations. In a joint research project on COA analysis between Army Research Laboratory and Ohio State University, Kaste et al [Kaste] reported that they used the multi-criterial decision tool developed at OSU to mine ARL combat simulation data in order to gain the battle-planning insights into understanding the COA space. The approach taken in this work is more related to data mining and visualization through user interaction to develop such insights. In a related work, Hillis et al [Hillis] reported an effort in surveying several existing tools for visualization for COA analysis including the OSU developed tools, and combined the existing tools together using coevolution and Pareto optimization for COA analysis.

Vakas et al [Vakas01] studied the situation assessment and COA selection using a Commander Model under the Joint Warfare System environment [JWARS01]. Fuzzy rules are used due to the typical fuzzy nature of the commander's intent, and users are provided with the ability to modify both the input parameters and the underlying rules. Zhang et al [Zhang02] reported a software system for COA development and analysis based on Colored Petri Nets [Jensen92]. The colored Petri Net model is used in this study to specify the execution and analysis of tasks in a COA.

Recently, COA analysis has been investigated in the context of real-time performance at The Air Force Research Laboratory. Gilmour et al [Gilmour05a, Gilmour05b] surveyed the current status of COA analysis and reported the approaches AFRL is taking on real-time COA analysis. Preliminary simulation results are reported using high performance computing facilities to achieve real-time COA analysis. Hanna et al [Hanna05] reported a data representation ontology and the related schema developed for the COA analysis at AFRL.

It is well-observed [Kyburg96] that in many real-world problems, the classic symbolic reasoning [Fikes03a, Fikes03b] may not work, and consequently, the research on uncertainty reasoning [Kyburg96, Harrington96] has received intensive attention.

4.0 CAFSIN Solution

Due to the great challenge of the semantic gap between the commander's intent and a COA, during the first phase of this investigation, we have made the following assumptions to simplify the solution:

1. The commander's intent is given at the tactical level. This allows a restrictive syntax to be used.
2. COA is also given in a lower, more specific level. This also allows a restrictive syntax to be used.
3. A domain ontology must be given.

Based on these assumptions, we propose the CAFSIN solution, standing for COA Analysis based on Fuzzified Semantic Inference. This solution models the determination of the alignment problem between a commander's intent and a COA as a fuzzified language matching problem. This is a general approach to COA analysis and reasoning because it addresses the uncertain and fuzzy nature of the problem using fuzzy logic analysis, and consequently, the solution leaves a user to define what is considered as a compliant COA or a diverting COA to allow user interaction. Even though CAFSIN is developed under the assumptions made above, it may also work when the assumptions are relaxed if reliable information extraction (IE) tools are available.

4.1 Ontology Construction

In order to facilitate the search in CAFSIN, when we construct the ontology, attention must be paid to the following issues.

1. Synonymy: all the synonyms are hard-wired together in a node in the ontology (e.g., “the Pentagon” and “DoD” are wired together as the same word and are represented as a single node).
2. Polysemy: words with different meanings in the ontology are represented in different nodes (e.g., “chair” as a department chair and “chair” as a piece of furniture are represented and located as separate words).
3. Special names: special names and phrases are coined as single words in the ontology (e.g., “WMD support system” as one word).

Given an ontology with these requirements satisfied, a standard hashing function may be used to directly identify a specific node in the ontology.

4.2 Fuzzified Word Similarity

Given two words *w1* and *w2* and an ontology Ψ, the similarity function f is defined as a Gaussian function:

$$f(w1, w2 \mid \Psi) = \frac{p}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\max\{d_{w1}, d_{w2}\}-1)^2}{2\sigma^2}} \qquad (1)$$

where d_{w1} and d_{w2} are the depths of *w1* and *w2* from a nearest common ancestor in Ψ; if they do not share a common ancestor, they are set as ∞; *p* here is the normalization factor; σ is the standard deviation.

Based on the definition of this fuzzified word similarity function, given an ontology, the similarity between two words depends on two things: (1) the relative depth difference between the two words in the ontology; and (2) the depth from the nearest common ancestor in the ontology.

Thus, two words have a strong similarity if they are synonyms, or siblings sharing a common parent, or one is a parent of the other. The similarity decreases if the depth difference between the two words increases in the same ontology tree; and/or their nearest common ancestor moves away. The similarity becomes 0 if the two words do not have a common ancestor, i.e., they are located in different ontology trees.

4.3 Language Models

Since we have assumed that the commander’s intent is represented as a tactical level command, and since at the tactical level, commands may be represented in a well-defined syntax, we use the following grammar as the language model for the commander’s intent:

$$T = \langle verb \rangle \ \langle noun \rangle + \qquad (2)$$

Similarly, a COA may be represented as a language sentence with the following grammar:

$$C = \{<verb>\ <attribute\ value>^*\}+ \qquad (3)$$

4.4 CAFSIN Similarity Function

Now we are ready to define the CAFSIN similarity function based on our CAFSIN design principle. Let $t \in T$, $t = v\ n^*$; let $c \in C$, $c = \{u\ m^*\}+$. Then the CAFSIN similarity function is defined as:

$$h(t, c \mid \Psi) = \alpha \Sigma_u f(v, u \mid \Psi) H(n^*, m^*) \qquad (4)$$

where $H(n^*,M^*)$ is a fuzzified maximum substring matching function between word string n^* and word string m^* using the fuzzified word similarity function $f(n,m| \Psi)$; α is a normalization factor.

Now the next question is how to compute the fuzzified maximum substring matching function H. It is well known that the substring matching problem is NP-complete, and an optimal solution may be found using dynamic programming. Since the actual t and c typically only have a very few words, the complexity is not an issue.

Assume that there are N words for the string n^*, and that there are M words for string m^*. Using the dynamic programming technique, we create a table of $H[N+1, M+1]$, and the table H is initialized as $H[0, j]=0$ for $j=1, \ldots , M+1$; and $H[i, 0]=0$ for $i=1, \ldots , N+1$. Thus, the rest of the entries in H are computed based on the following recurrence:

$$H[i, j] = \begin{cases} f(n[i], m[j]), \ f(n[i], m[j]) > \max(H[i-1, j], H[i, j-1]) \\ \max(H[i-1, j], H[i, j-1]), otherwise \end{cases} \qquad (5)$$

5.0 Proof-of-Concept Testing

The CAFSIN method has been tested with a hand-crafted ontology. Figures 5 and 6 show part of the ontology.

As a first example, assume that we have a commander's intent as "disrupt enemy's WMD support system" and a COA as "engage FA-18 target B13". After the standard text processing, we have the two strings for the commander's intent and the COA as represented in Figure 4.

t = {<disrupt>,(<enemy>,<WMD support system>)}
c = {<engage>,(<FA-18>,<target>,<B13>)}

Figure 4: The commander's intent and the COA strings after the text processing

Based on the ontology of Figures 5 and 6, we have obtained the distances between the relevant word pairs as in Table 1.

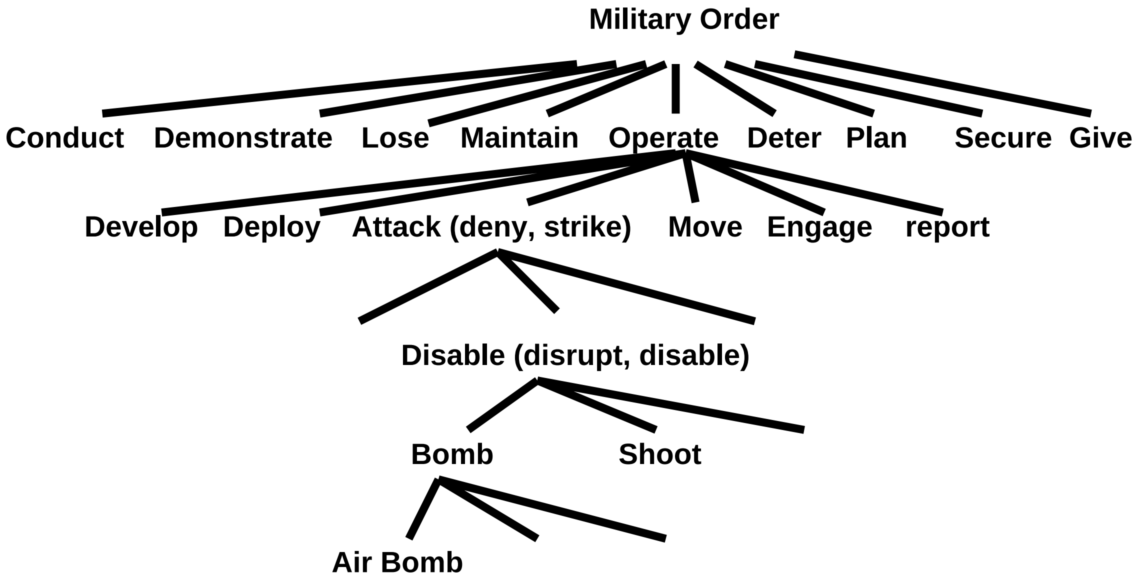

Figure 5: Part of the ontology used in the preliminary testing.

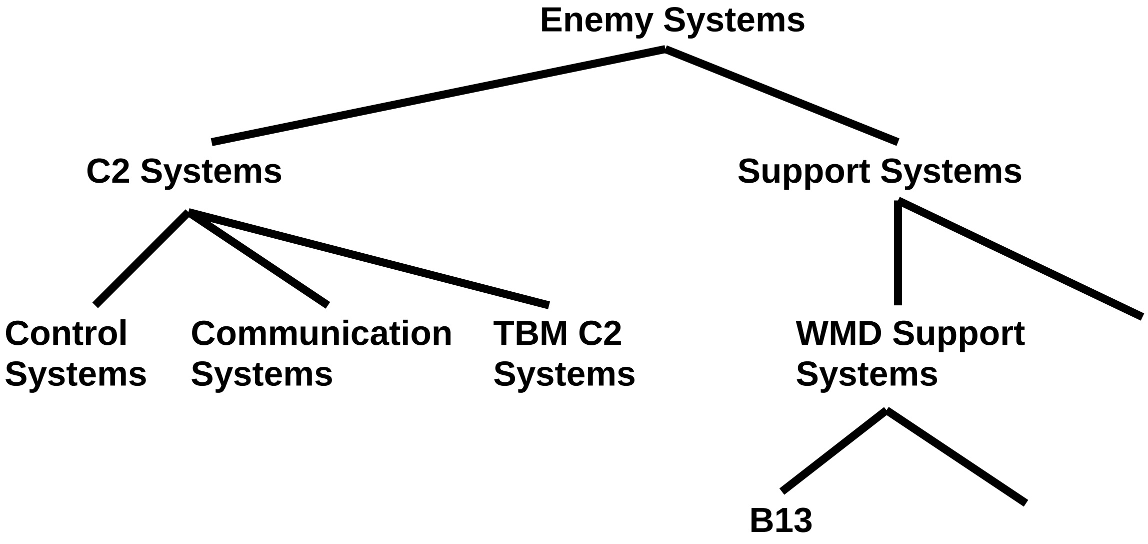

Figure 6: Another part of the ontology used in the preliminary testing.

Assuming the parameters of σ=1, $p=\sqrt{2\pi}$, α=1, from Eq. 1, we compute the fuzzified similarity values for the relevant word pairs as reported in Table 2. Finally, the *H* function is computed using dynamic programming based on Eq. 5 as denoted in Table 3, and the final similarity function value between the pair of commander's intent and the COA is determined based on Eq. 4 as 0.607.

Table 1: Ontology distances between the words in the example.

d<disrupt> = 2	d<engage> = 1
d<enemy> = ∞	d<FA-18> = ∞
d<enemy> = ∞	d<target> = ∞
d<enemy> = ∞	d<B13> = ∞
d<WMD support system> = ∞	d<FA-18> = ∞
d<WMD support system> = ∞	d<target> = ∞
d<WMD support system> = 0	d<B13> = 1

Table 2: Similarity values between the words in the example.

f(<disrupt>, <engage>) = 0.607
f(<enemy>, <FA-18>) = 0
f(<enemy>, <target>) = 0
f(<enemy>, <B13>) = 0
f(<WMD support system>, <FA-18>) = 0
f(<WMD support system>, <target>) = 0
f(<WMD support system>, <B13>) = 1

Table 3: The dynamic programming table for computing *H* in the example.

		enemy	WMD support system
	0	0	0
FA-18	0	0	0
target	0	0	0
B13	0	0	1

As another example, we have the same commander's intent as in the first example, but the COA has one more action "bomb target B13" in addition to the original action in the first example. After standard text processing we have the strings specified in Figure 7.

t = {<disrupt>,(<enemy>,<WMD support system>)}
c ={<engage>,(<FA-18>,<target>,<B13>);
<bomb>,(<target>,<B13>)}

Figure 7: Another example of commander's intent and COA specified as strings after text processing.

The similarity computation for the pair between the commander's intent and the first action of the COA is identical to the one computed in the first example, and the similarity value is 0.607. In order to compute the similarity for the pair between the same commander's intent and the second part of the COA, we first determine the ontology distance based on the ontology described in Figures 5 and 6, as reported in Table 4. The word pair similarity values are then computed using Eq. 1 and shown in Table 5, and finally the H function is computed using dynamic programming based on Eq. 5 shown and in Table 6. Thus, the final similarity value between the commander's intent and the COA is summation of the two parts which becomes 1.607. Clearly, due to the addition of the second part in the COA, the similarity of this COA to the commander's intent is much higher than the one in the first example, which indicates that the second COA is more in alignment with the commander's intent than the first COA.

Table 4: Ontology distance between the words for the second pair in the second example.

d<disrupt> = 0	d<bomb> = 1
d<enemy> = ∞	d<target> = ∞
d<enemy> = ∞	d<B13> = ∞
d<WMD support system> = ∞	d<target> = ∞
d<WMD support system> = 0	d<B13> = 1

Table 5: Similarity values between the words for the second pair in the second example.

f(<disrupt>, <bomb>) = 1
f(<enemy>, <target>) = 0
f(<enemy>, <B13>) = 0
f(<WMD support system>, <target>) = 0
f(<WMD support system>, <B13>) = 1

Table 6: Dynamic programming table for computing *H* for the second pair in the second example.

		enemy	WMD support system
	0	0	0
target	0	0	0
B13	0	0	1

As yet another example, let's stay with the same commander's intent used in the previous two example but the COA now becomes "Lose target B13". Is this COA in alignment with the commander's intent? After the standard text processing, we have the strings specified in Figure 8.

t = {<disrupt>,(<enemy>,<WMD support system>)}
c = {<lose>,(<target>,<B13>)}

Figure 8: The strings specified for the commander's intent and the COA after standard text processing in the third example.

Now the same CAFSIN method is applied to this example reported in Tables 7 – 9, resulting in the final similarity value between the commander's intent and the COA as 0.135 for this example, assuming the same parameters are used as in the first two examples. This low similarity value indicates that the COA in this example is not quite in alignment with the commander's intent.

Table 7: Ontology distance between the words in the third example.

d<disrupt> =	d<lose> = 1
d<enemy> = ∞	d<target> = ∞
d<enemy> = ∞	d<B13> = ∞
d<WMD support system> = ∞	d<target> = ∞
d<WMD support system> = 0	d<B13> = 1

Table 8: Similarity values between the words in the third example.

f(<disrupt>, <bomb>) = 0.135
f(<enemy>, <target>) = 0
f(<enemy>, <B13>) = 0
f(<WMD support system>, <target>) = 0
f(<WMD support system>, <B13>) = 1

Table 9: Dynamic programming table for computing *H* in the third example.

		enemy	WMD support system
	0	0	0
target	0	0	0
B13	0	0	1

The above three examples demonstrate the effectiveness of using CAFSIN method to determine the alignment between a pair of given commander's intent and COA, and showcases the proof of the concept for CAFSIN method. Using the ontology we have hand-crafted shown in Figures 5 and 6, we have tested numerous examples for different commander's intent and COAs and the experiments all show that CAFSIN is a very effective method to serve the purpose of determining whether a COA is in alignment with a commander's intent, and in case not, how much that COA diverts the commander's intent.

6.0 Conclusion and Future Work

In this summer research, we have reviewed the relevant literature regarding the problem of determining the alignment of a course of action with a commander's intent, and have found that the solution to this problem must address explicitly the uncertainty and fuzziness issues in the reasoning process. Consequently, we have concluded that the classic symbolic reasoning does not work. Instead, we have proposed a specific but general solution to the problem that is based on fuzzified semantic inference called CAFSIN. We have demonstrated that CAFSIN is an effective method to solve for the problem through proof-of-concept testing.

CAFSIN has the following advantages:

- It is independent of ontology; we can apply it to any domain to solve for the problem.
- It is independent of specific words; the similarity between the words is only dependent on the relative locations between the words in ontology.
- It is always relative; this allows users interaction based on their experience and expertise.

The future work includes:

1. Large scale evaluation.

Large scale evaluation is necessary in order to actually turn the CAFSIN method into a real world technology. However, there are several issues that we must address before we are able to conduct large scale evaluations. The first is the construction of an ontology for a specific application domain. How to construct such a reasonably large ontology is always a question. We may use the existing general domain ontology such as Wordnet. But how to tailor the general ontology to the specific application domain also becomes an issue. Finally, even if we have an ontology and a data set ready for the evaluations, how to define the evaluation metrics is another issue.

2. Relaxing the assumptions.

There are two directions that we can go for relaxing the current assumptions of CAFSIN. The first one is to apply CAFSIN to a higher level of commander's intent and COA. In this case we need tools for natural language processing. If we have reliable IE tools available, the direct extension of CAFSIN to this scenario is not a problem; the problem now becomes how to interface the IE tools into the CAFSIN method. The second direction is to relax the syntax of COA to incorporate constraints into the COA. This involves revising the CAFSIN strings matching function.

3. Improving the computation complexity.

There are two bottlenecks to the complexity of the CAFSIN method. The first is the fuzzified substring matching to compute the *H* function. We claim that typically the strings are not very long and so the complexity would not be a problem. However, in case the strings become very long, we may need to add heuristics into the matching to expedite the substring search. The second bottleneck is the ontology tree search for identifying the correct location of the word. Given the typical scenario that the words in a given COA may be located "close enough" in the ontology, we may be able to add some locality analysis into the ontology tree search for reducing the tree traversal search time.

7.0 Acknowledgements

I would like to thank my mentor, Duane Gilmour, for his advice, insights, and strong support to this project, as well as accommodating me in a flexible schedule to take care of the work at my university, the service in my profession, and my family. I would also like to thank John Graniero, Barry McKinney, and Anna Lemaire for their support and advice that made my visiting tour a pleasant and enjoyable stay and made this stay productive.

8.0 References

[Chandrasekaran99] B. Chandrasekaran and J.R. Josephson, Cognitive modeling for simulation goals: a research strategy for computer-generated forces, *Proceedings of computer generated forces*, 1999

[Fikes03a] R. Fikes, J. Jenkins, and Q. Zhou, Including domain-specific reasoners with reusable ontologies, *Proceedings of the international conference on information and knowledge engineering*, 2003

[Fikes03b] R. Fikes, G. Frank, and J. Jenkins, JTP: a system architecture and component library for hybrid reasoning, *Proceedings of the 7th multi-conference on systems, cybernetics, and informatics*, 2003

[Gilmour05a] D.A. Gilmour, J.P. Hanna, W.A. Koziarz, W.E. McKeever, and M.J. Walter, High-performance computing for command and control real-time decision support, *Horizons*, 6(1), Feb., 2005

[Gilmour05b] D.A. Gilmour, J.P. Hanna, W.E. McKeever, and M.J. Walter, Real-time course of action analysis, *Proceedings of the 10th International Command and Control Research and Technology Symposium*, 2005

[Harrington96] R.A. Harrington, S. Banks, and E. Santos, The pski intelligent user interface, *TechReport*, department of electrical and computer engineering, AFIT, 1996

[Hillis] D. Hillis, M. Barnes, L. Suantak, J. Schlabach, B. Chandrasekaran, J.R. Josephson, and M. Carroll, Collaborative visualization tools for courses of action (COA) in anti-terrorist operations: combining coevolution and pareto optimization, Report available at AFRL

[Jensen92] K. Jensen, *Coloured Petri Net: Basic Concepts, Analysis Methods and Practical Use*, Volumes 1 – 3, Springer Verlag, 1992 – 1997

[JWARS01] Joint Warfare System Commander Behavior Model, *Phase I TechReport*, Joint Warfare System Office, CACI, 2001

[Kaste] R. Kaste, J. O'May, E. Heilman, B. Chandrasekaran and J. Josephson, From simulation to insights: experiments in the use of a multi-criterial viewer to develop understanding of the COA space, Report available at AFRL

[Kyburg96] H.E. Kyburg, Jr. Uncertain inferences and uncertain conclusions. *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, 1996

[Vakas01] D. Vakas, J. Prince, H.R. Blacksten, and C. Burdick, Commander behavior and course of action selection in JWARS, *Proceedings of the 2001 winter simulation conference*, 2001

[Zhang02] L. Zhang, L.M. Kristensen, C. Janczura, G. Gallasch, and J. Billington, A coloured Petri net based tool for course of action development and analysis, *Proceedings of the workshop on formal methods applied to defence systems*, 2002